**Epic: Develop and Deploy a GenAI-Powered Microservice with FastAPI**

**User Story 1: Microservice Development with FastAPI**

* **Title**: Develop a GenAI-Powered Microservice using FastAPI
* **Description**: Create a microservice using FastAPI that integrates a generative AI model to perform tasks such as text generation
* **Acceptance Criteria**:
* The microservice should be implemented using FastAPI.
* It should perform at text generation task using a OpenAI.
* The source code should be available in a GitHub repository.

**User Story 2: Containerization and Deployment**

* **Title**: Containerize and Deploy Microservice to DockerHub
* **Description**: Containerize the microservice and deploy it to DockerHub, ensuring it is ready for cloud deployment.
* **Acceptance Criteria**:
* A Dockerfile should be created for the FastAPI microservice.
* The Docker image should be pushed to DockerHub.
* Kubernetes deployment scripts should be prepared for cloud deployment.

**User Story 3: CI/CD Pipeline with GitHub Actions**

* **Title**: Implement CI/CD Pipeline using GitHub Actions
* **Description**: Set up a CI/CD pipeline using GitHub Actions to automate the build, test, and deployment processes.
* **Acceptance Criteria**:
* GitHub Actions should be configured to automate testing and deployment.
* The pipeline should build the Docker image and push it to DockerHub.
* Automated tests should be run as part of the CI/CD process.

**User Story 4: Documentation and Quality Assurance**

* **Title**: Document and Ensure Quality of the Microservice
* **Description**: As a user, I want clear documentation and assurance of quality for the microservice.
* **Acceptance Criteria**:
  + A README file should be included with setup and usage instructions.
  + Unit tests, integration tests, and performance tests should be implemented.
  + Security best practices should be followed, including code reviews.

**Tasks for Each User Story**

**User Story 1: Microservice Development with FastAPI**

* **Title: Develop a GenAI-Powered Microservice using FastAPI**

**Tasks:**

1. **Task 1.1: Set up FastAPI Project**

* Initialize a new FastAPI project.
* Set up a virtual environment and install necessary dependencies (e.g., FastAPI, Uvicorn).
* Create a basic project structure with directories for routes, models, and services.

1. **Task 1.2: Integrate OpenAI API**

* Implement a service layer to interact with the OpenAI model.
* Create utility functions for model input/output processing.

1. **Task 1.3: Implement API Endpoints**

* Design API endpoints for tasks such as text generation
* Implement the endpoints using FastAPI, ensuring proper request validation and error handling.
* Write documentation for each endpoint using FastAPI's built-in documentation features.

1. **Task 1.4: Test API Functionality**

* Write basic tests to ensure each endpoint functions correctly.
* Use tools like pytest to automate testing.

**User Story 2: Containerization and Deployment**

* **Title: Containerize and Deploy Microservice to DockerHub**

**Tasks:**

1. **Task 2.1: Write Dockerfile**

* Create a Dockerfile to containerize the FastAPI application.
* Ensure the Dockerfile includes all necessary dependencies and optimizes for size and performance.
* Test the Dockerfile locally to ensure it builds correctly.

1. **Task 2.2: Build and Test Docker Image**

* Build the Docker image locally using the Dockerfile.
* Run the Docker container locally to test the application in a containerized environment.
* Debug any issues that arise during containerization.

1. **Task 2.3: Push Docker Image to DockerHub**

* Create a DockerHub account and repository if not already available.
* Tag the Docker image appropriately for versioning.
* Push the Docker image to DockerHub.

1. **Task 2.4: Prepare Kubernetes Deployment Scripts**

* Write Kubernetes deployment and service YAML files.
* Ensure configurations are cloud-agnostic to facilitate deployment on any cloud platform.

**User Story 3: CI/CD Pipeline with GitHub Actions**

* **Title: Implement CI/CD Pipeline using GitHub Actions**

**Tasks:**

1. **Task 3.1: Configure GitHub Actions Workflow**

* Create a .github/workflows directory in the repository.
* Write a GitHub Actions YML file to define the CI/CD pipeline.
* Include steps for checking out code, setting up Python, and installing dependencies.

1. **Task 3.2: Automate Testing**

* Add steps in the GitHub Actions workflow to run unit and integration tests.
* Ensure test results are reported, and any failures halt the pipeline.

1. **Task 3.3: Automate Docker Build and Push**

* Add steps to build the Docker image as part of the CI/CD pipeline.
* Configure GitHub Actions to authenticate with Docker Hub and push the image.

1. **Task 3.4: Implement Deployment Automation**

* Add steps to deploy the application to a Kubernetes cluster.
* Use secrets management to handle sensitive information like API keys and Docker Hub credentials.

**User Story 4: Documentation and Quality Assurance**

* **Title: Document and Ensure Quality of the Microservice**

**Tasks:**

1. **Task 4.1: Write Comprehensive README**

* Document the setup process, including environment setup and dependency installation.
* Provide instructions for running the application locally and in a container.
* Include API documentation and usage examples.

1. **Task 4.2: Develop Unit and Integration Tests**

* Write unit tests for individual components and functions.
* Develop integration tests to ensure components work together as expected.
* Use a testing framework like pytest for automation**.**

1. **Task 4.3: Implement Performance Testing**

* Identify key performance metrics and scenarios.
* Use tools like Locust or Apache JMeter to simulate load and measure performance.
* Document performance benchmarks and any optimizations made.

1. **Task 4.4: Conduct Code Reviews and Security Assessments**

* Set up a code review process using GitHub pull requests.
* Use static analysis tools to identify potential security vulnerabilities.
* Conduct a vulnerability assessment and document findings and mitigations.